

Soluzioni Esercizi

Reti Combinatorie

Reti Logiche T

Ingegneria Informatica

Esercizio 1 – valutazione delle funzioni

Caso $x_4x_3 = 00 \rightarrow U = x_0x_1x_2 + x_0'x_1'x_2$

$x_1x_0 \backslash x_2$	00	01	11	10
0	0	0	0	0
1	1	0	1	0

Caso $x_4x_3 = 01 \rightarrow U = x_0 + x_1 + x_2$

$x_1x_0 \backslash x_2$	00	01	11	10
0	0	1	1	1
1	1	1	1	1

Caso $x_4x_3 = 10 \rightarrow U = (x_0 \oplus x_1 \oplus x_2) + x_2'$

$x_1x_0 \backslash x_2$	00	01	11	10
0	1	1	1	1
1	1	0	1	0

Caso $x_4x_3 = 11 \rightarrow U = (x_0x_1x_2)'$

$x_1x_0 \backslash x_2$	00	01	11	10
0	1	1	1	1
1	1	1	0	1

Esercizio 1 – mappe complete

Caso $x_4x_3 = 00 \rightarrow U = x_0x_1x_2 + x_0'x_1'x_2$

x_1x_0 x_2	00	01	11	10
0	0	0	0	0
1	1	0	1	0

x_1x_0 x_3x_2	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	1	1	1	1
10	0	1	1	1

Caso $x_4x_3 = 01 \rightarrow U = x_0 + x_1 + x_2$

x_1x_0 x_2	00	01	11	10
0	0	1	1	1
1	1	1	1	1

$x_4 = 0$

Esercizio 1 – mappe complete

Caso $x_4x_3 = 10 \rightarrow U = (x_0 \oplus x_1 \oplus x_2) + x_2'$

$x_1x_0 \backslash x_2$	00	01	11	10
0	1	1	1	1
1	1	0	1	0

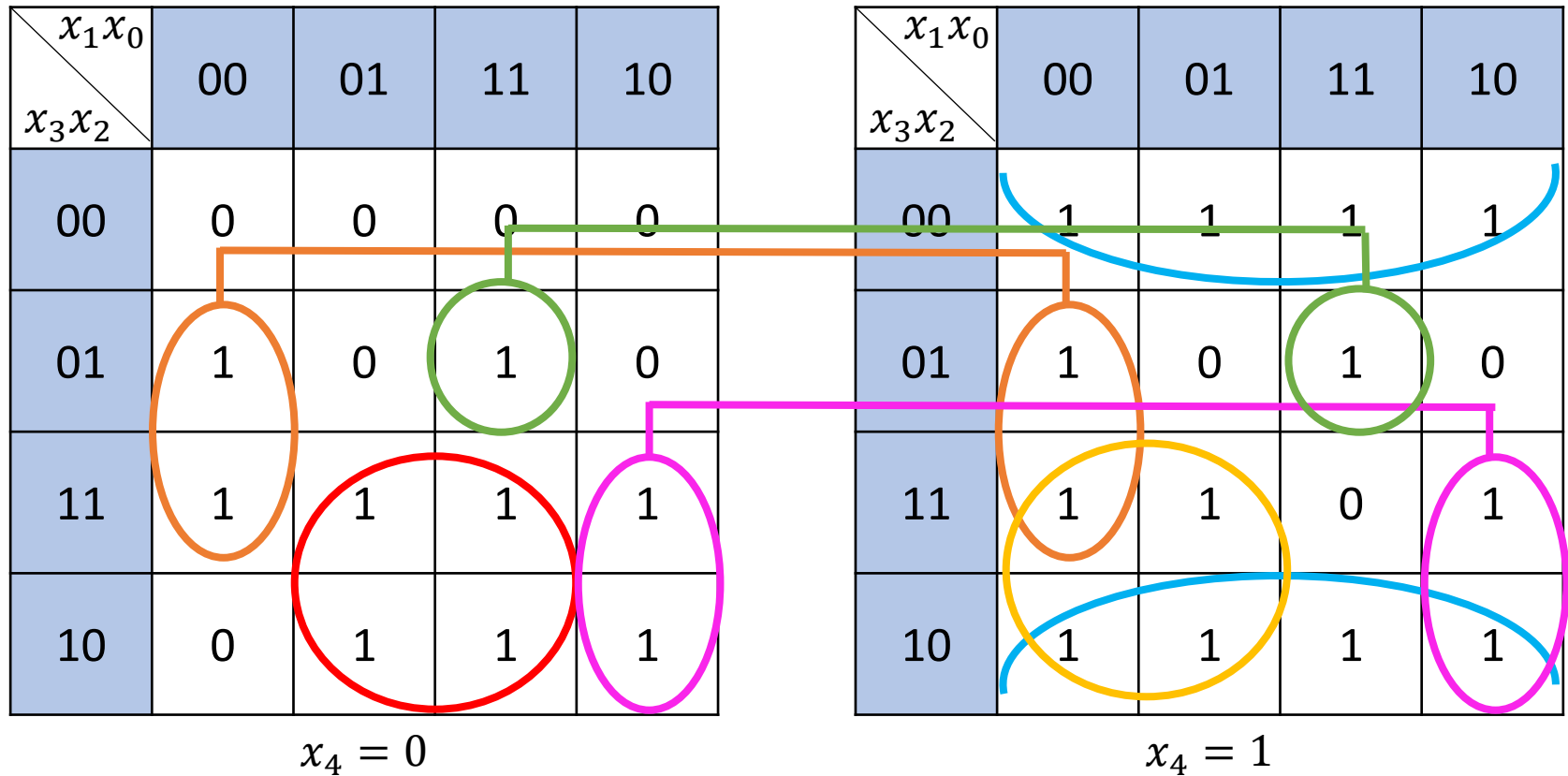
$x_1x_0 \backslash x_3x_2$	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	1	1	0	1
10	1	1	1	1

Caso $x_4x_3 = 11 \rightarrow U = (x_0x_1x_2)'$

$x_1x_0 \backslash x_2$	00	01	11	10
0	1	1	1	1
1	1	1	0	1

$x_4 = 1$

Esercizio 1 – sintesi SP



$$U = x_2x_1'x_0' + x_3'x_2x_1x_0 + x_4'x_3x_0 + x_3x_1x_0' + x_4x_3x_1' + x_4x_2'$$

Esercizio 1 – sintesi PS

x_1x_0 x_3x_2	00	01	11	10
00	0	0	0	0
01	1	0	1	0
11	1	1	1	1
10	0	1	1	1

$x_4 = 0$

x_1x_0 x_3x_2	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	1	1	0	1
10	1	1	1	1

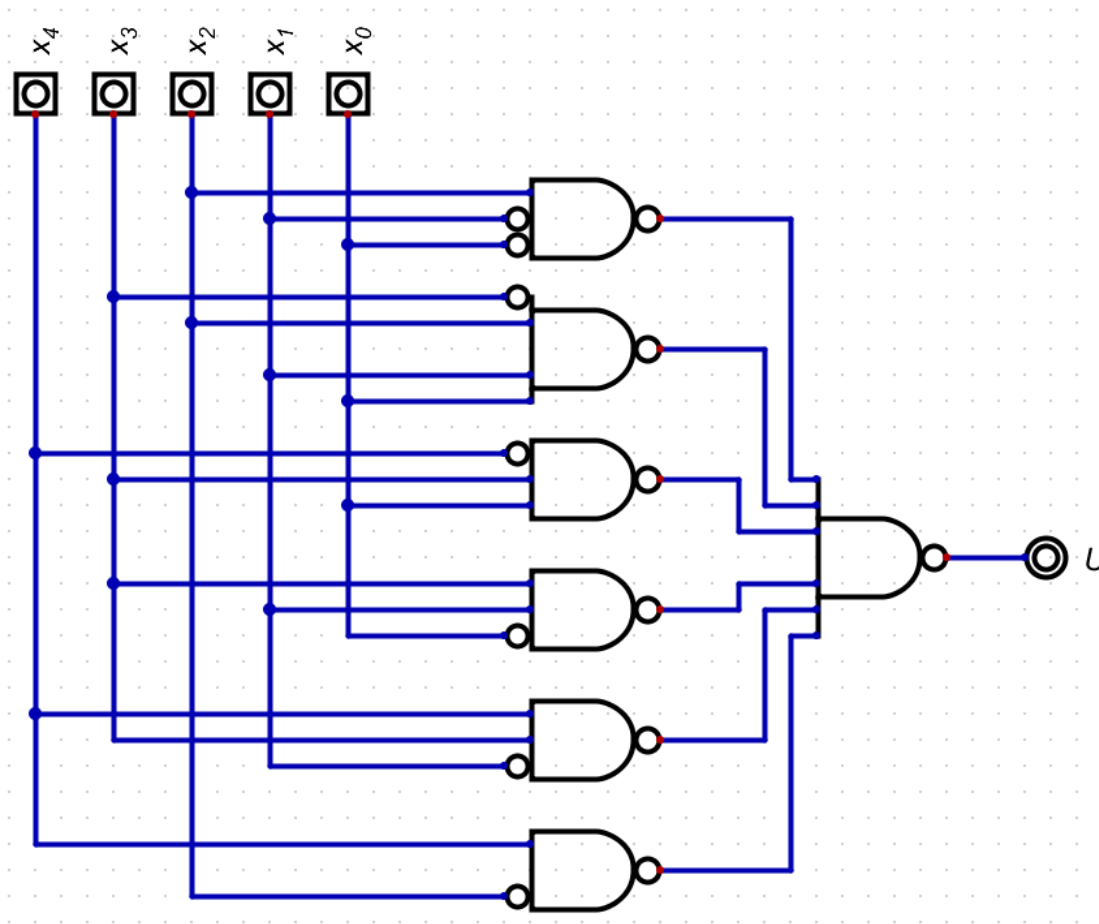
$x_4 = 1$

$$U = (x_4 + x_3 + x_2)(x_3 + x'_2 + x_1 + x'_0)(x_3 + x'_2 + x'_1 + x_0) \\ (x_4 + x_2 + x_1 + x_0)(x'_4 + x'_3 + x'_2 + x'_1 + x'_0)$$

Esercizio 1 – circuito a NAND

$$U = x_2 x_1' x_0' + x_3' x_2 x_1 x_0 + x_4' x_3 x_0 + x_3 x_1 x_0' + x_4 x_3 x_1' + x_4 x_2'$$

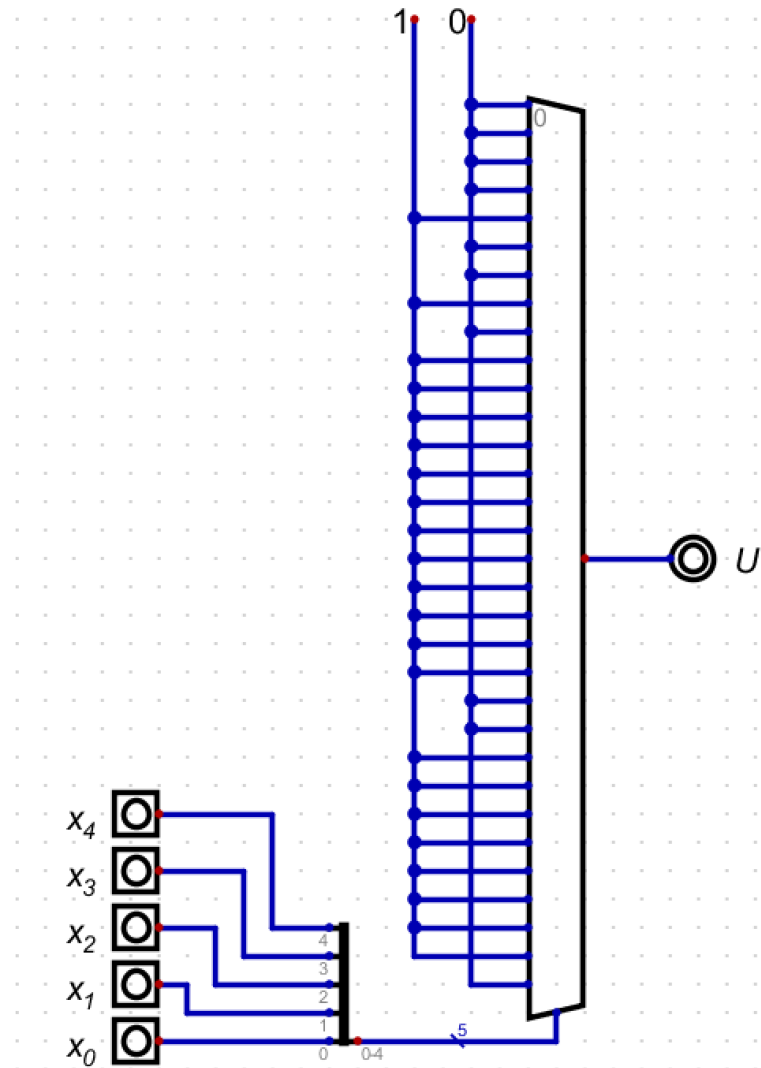
$$U = (x_2 \uparrow x_1' \uparrow x_0') \uparrow (x_3' \uparrow x_2 \uparrow x_1 \uparrow x_0) \uparrow (x_4' \uparrow x_3 \uparrow x_0) \uparrow (x_3 \uparrow x_1 \uparrow x_0') \uparrow (x_4 \uparrow x_3 \uparrow x_1') \uparrow (x_4 \uparrow x_2')$$



Esercizio 1 – mux a 5 bit di indirizzo

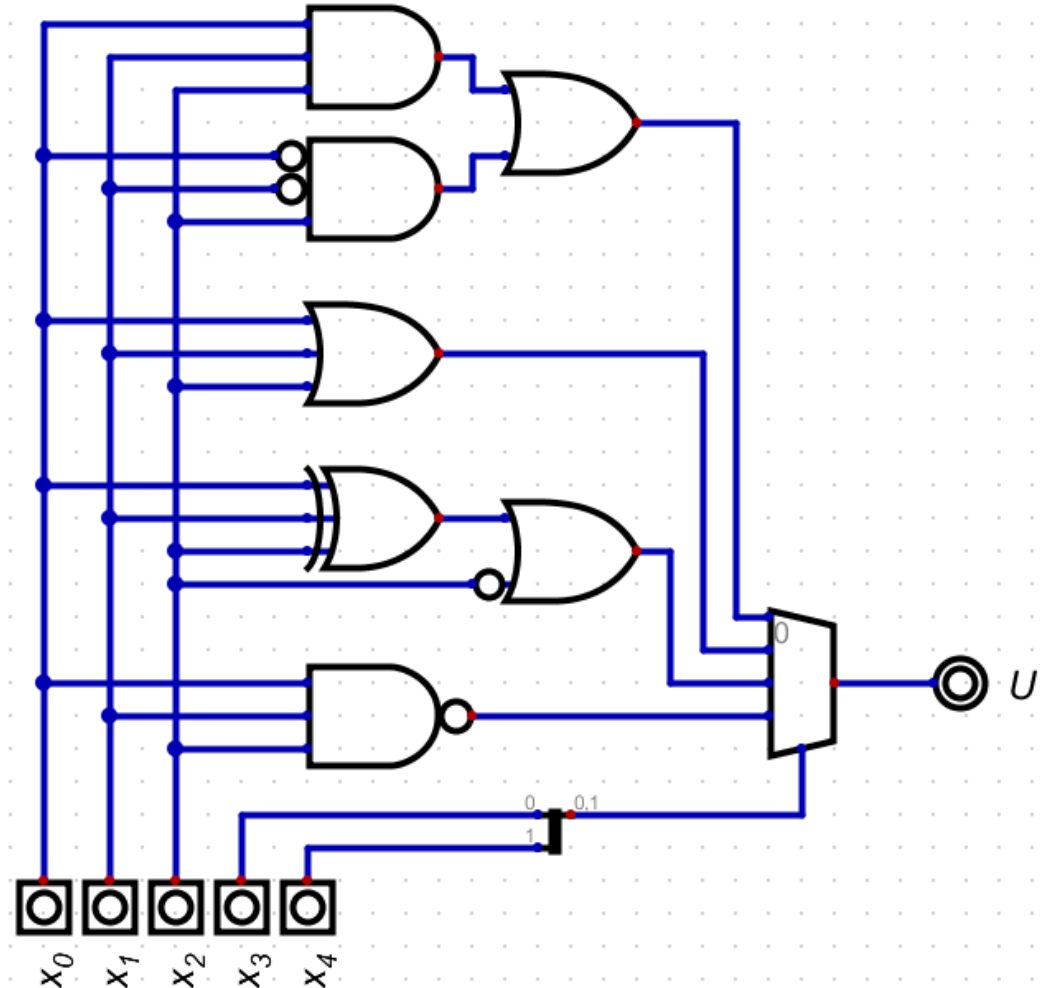
$x_4x_3x_2x_1x_0$	U
00000	0
00001	0
00010	0
00011	0
00100	1
00101	0
00110	0
00111	1
01000	0
01001	1
01010	1
01011	1
01100	1
01101	1
01110	1
01111	1

$x_4x_3x_2x_1x_0$	U
10000	1
10001	1
10010	1
10011	1
10100	1
10101	0
10110	0
10111	1
11000	1
11001	1
11010	1
11011	1
11100	1
11101	1
11110	1
11111	0



Esercizio 1 – mux a 2 bit di indirizzo

La soluzione con mux a 2 bit di indirizzo si ricava direttamente dalla tabella delle verità riportata nel testo dell'esercizio.



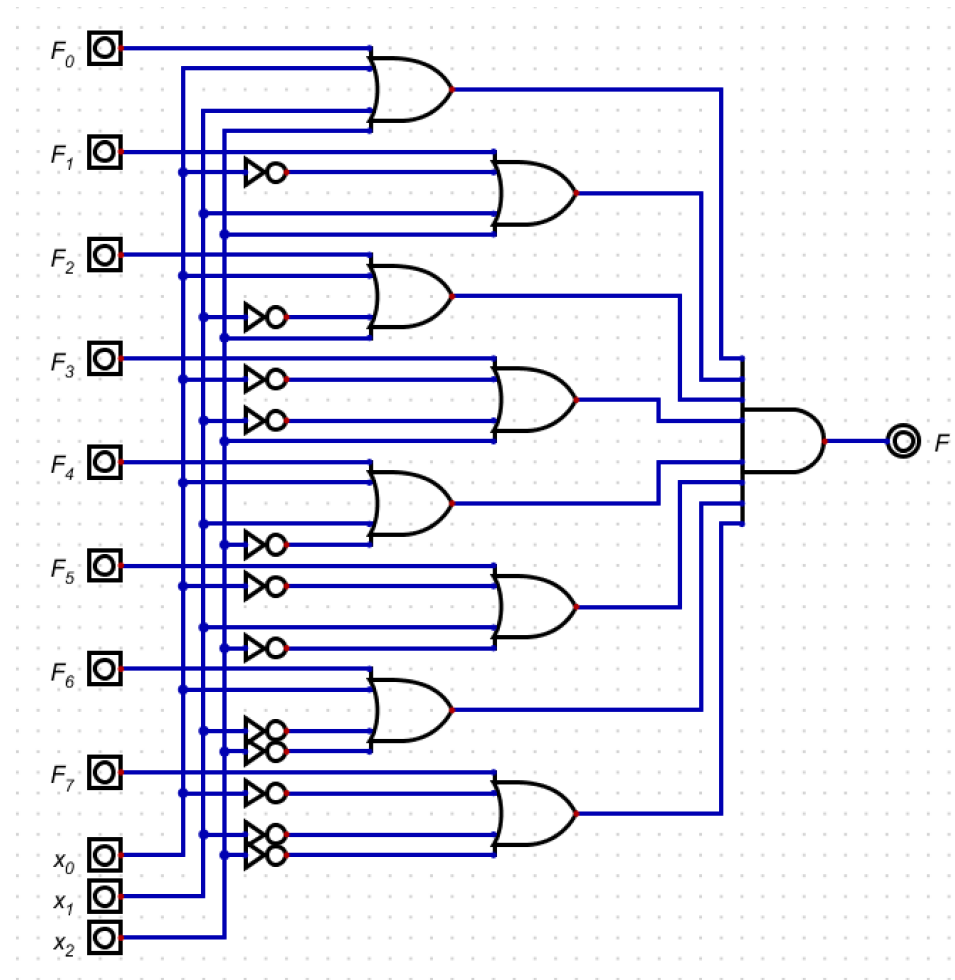
Esercizio 2 – Espressione generale PS

$$F(x_1, x_2, \dots, x_n) = \prod_{i=0}^{2^n-1} (M(i) + F(i))$$

$M(i)$: maxtermine di n bit

$F(i)$: valore della funzione relativo alla configurazione per cui $M(i) = 0$

Il numero di interconnessioni è $2^n(n+1) + 2^n$, se si dispone di segnali in forma vera e negata, altrimenti $n + 2^n(n+1) + 2^n$ se occorre considerare anche i NOT per ottenere i segnali in forma negata.



Esercizio 3 – sintesi a NOR

$$Z = ((a + b) \cdot (c + d) + a \cdot (b + c')) \cdot (a + d')$$

Esplicito le parentesi mancanti:

$$Z = (((a + b) \cdot (c + d)) + (a \cdot (b + c'))) \cdot (a + d')$$

Converto in sintesi a NOR l'AND più esterno:

$$Z = (((a + b) \cdot (c + d)) + (a \cdot (b + c'))) \downarrow (a + d')$$

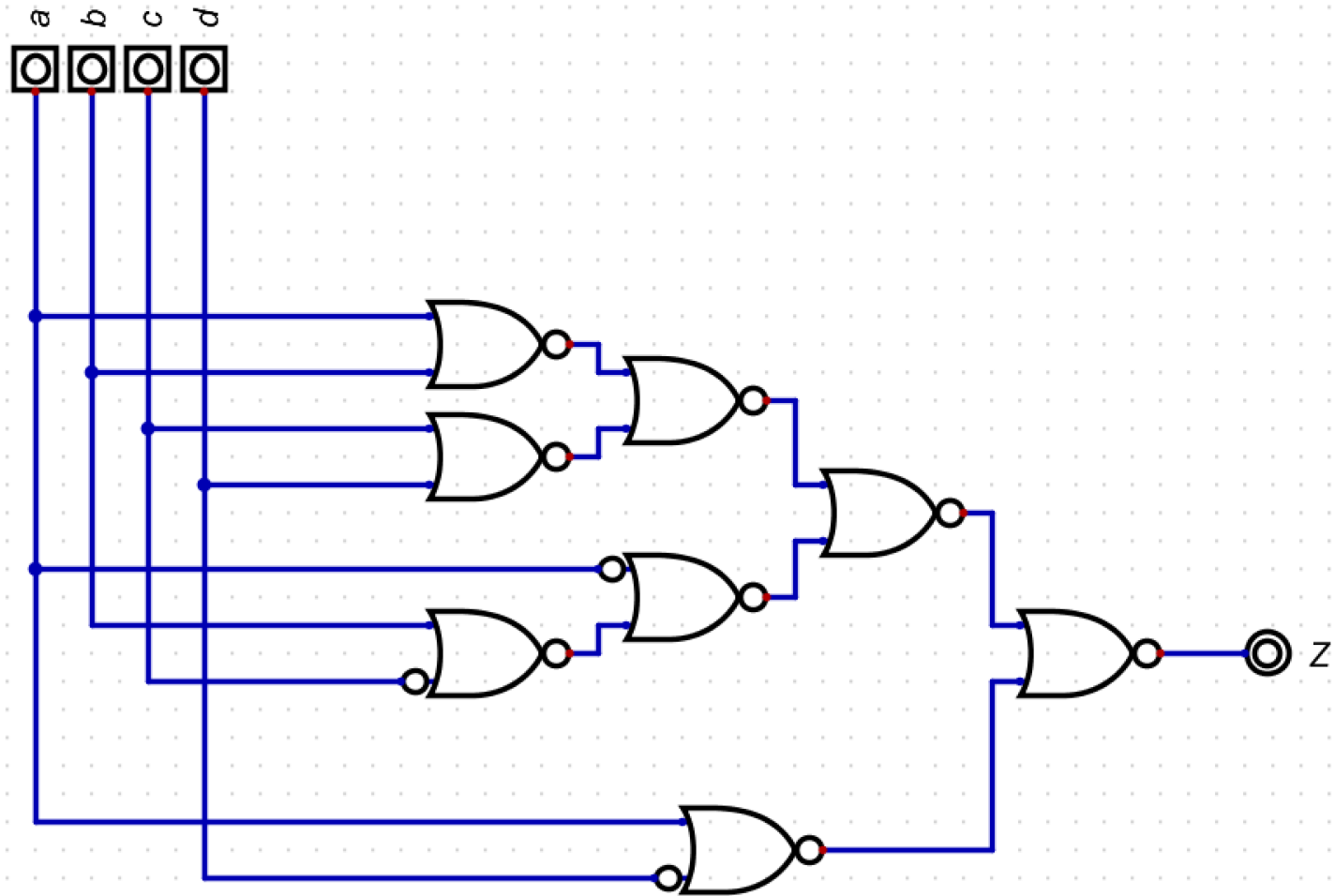
Converto in sintesi a NOR i due AND dentro al primo termine:

$$Z = (((a + b) \downarrow (c + d)) + (a' \downarrow (b + c'))) \downarrow (a + d')$$

Converto in sintesi a NOR gli OR

$$Z = (((a \downarrow b) \downarrow (c \downarrow d)) \downarrow (a' \downarrow (b \downarrow c'))) \downarrow (a \downarrow d')$$

Esercizio 3 – sintesi a NOR



Esercizio 3 – sintesi SP

$$Z = ((a + b) \cdot (c + d) + a \cdot (b + c')) \cdot (a + d')$$

Espansione della variabile a :

$$Z = a' \cdot Z(0, b, c, d) + a \cdot Z(1, b, c, d)$$

$$Z = a' \cdot ((\cancel{0} + b) \cdot (c + d) + \cancel{0} \cdot (\cancel{b + c'})) \cdot (\cancel{0} + d') + a \cdot ((\cancel{1} + b) \cdot (c + d) + \cancel{1} \cdot (b + c')) \cdot (\cancel{1} + d')$$

$$Z = \overset{E2}{a' \cdot (b \cdot (c + d)) \cdot d'} + a \cdot \overset{E2}{((c + d) + (b + c'))}$$

$$Z = a' \cdot b \cdot (c + d) \cdot d' + a \cdot \overset{E8}{(c + d + b + c')}$$

$$Z = a' \cdot b \cdot (c + d) \cdot d' + a \cdot \overset{E6 + E5}{(b + d + 1)}$$

$$Z = \overset{E4}{a' \cdot b \cdot (c + d) \cdot d'} + a$$

$$Z = a' \cdot b \cdot c \cdot d' + \overset{E8 + E6 + E5}{a' \cdot b \cdot d \cdot d'} + a$$

$$Z = \textcolor{violet}{a' b c d'} + \textcolor{green}{a}$$

→

$ab \backslash cd$	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	1	1	1
10	1	1	1	1

SINTESI
OTTIMA

$ab \backslash cd$	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	1	1	1	1
10	1	1	1	1

$$Z = \textcolor{violet}{b c d'} + \textcolor{green}{a}$$

Esercizio 4

A 11010 +

B 00100 =

S 11110

Conversione addendi in base 10:

$$11010 \rightarrow -2^4 + 2^3 + 2^1 = -16 + 8 + 2 = -6$$

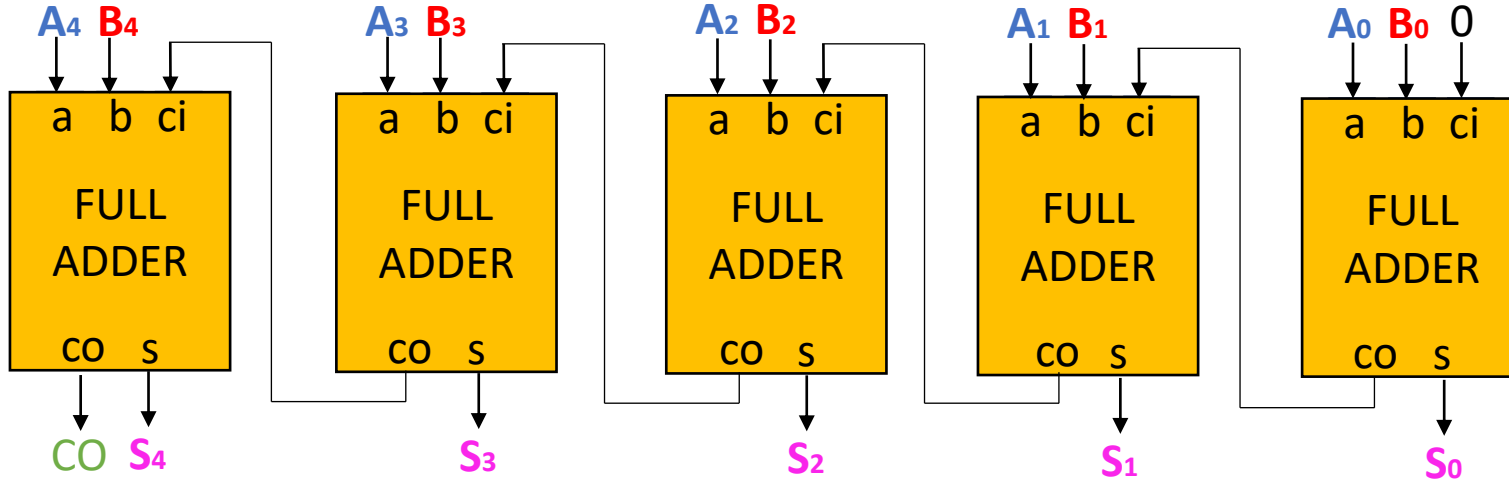
$$00100 \rightarrow 2^2 = 4$$

Somma in base 10:

$$-6 + 4 = -2$$

Conversione risultato in base 10:

$$11110 \rightarrow -2^4 + 2^3 + 2^2 + 2^1 = -16 + 8 + 4 + 2 = -2$$



5 Full Adder in cascata, ritardo massimo
(necessario per avere S_4 valido) pari a 50 ns

Esercizio 5 – sintesi SP

$de \backslash bc$	00	01	11	10
00	-	1	0	1
01	-	1	-	0
11	0	0	0	0
10	-	0	-	1

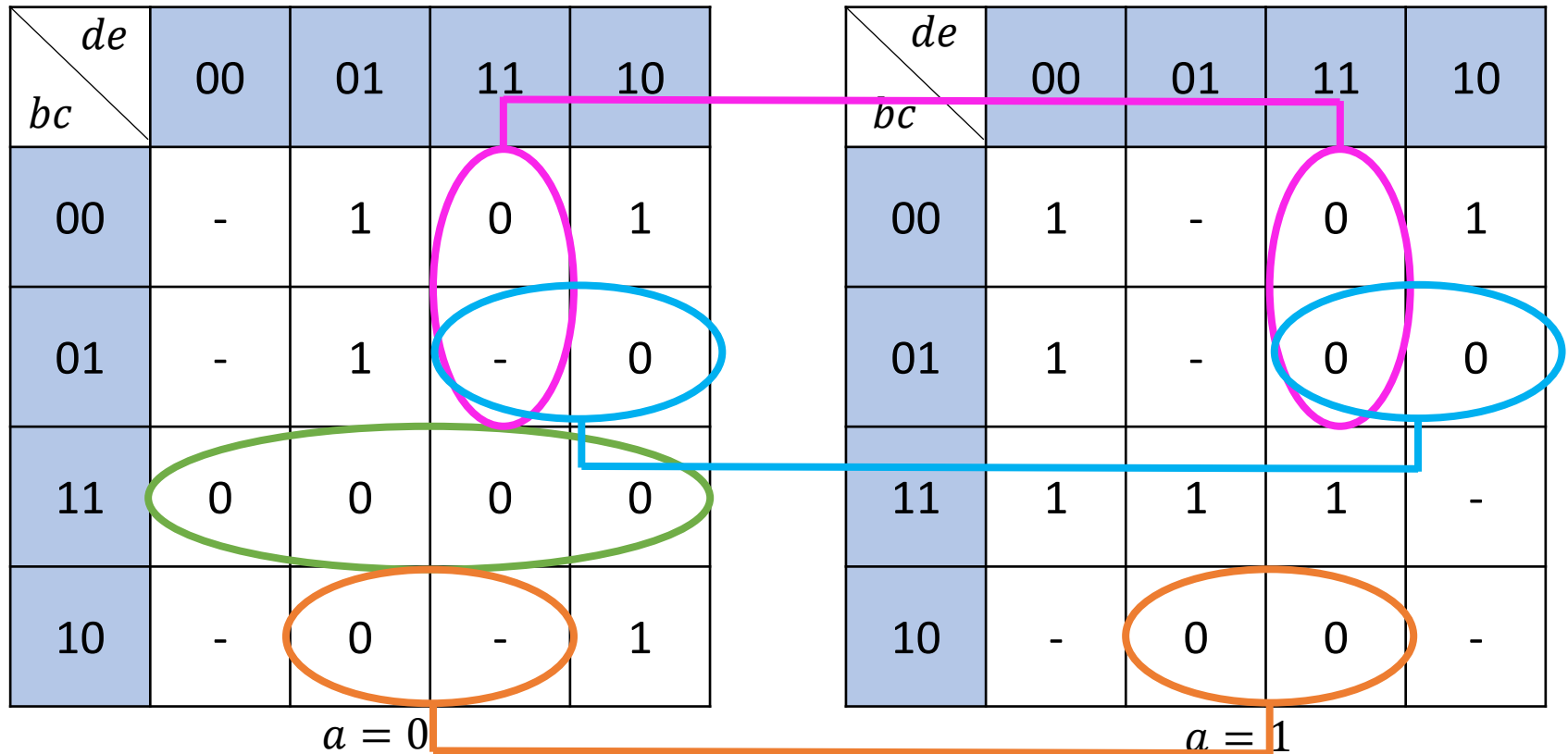
$a = 0$

$de \backslash bc$	00	01	11	10
00	1	-	0	1
01	1	-	0	0
11	1	1	1	-
10	-	0	0	-

$a = 1$

$$Z = b'd' + abc + c'e'$$

Esercizio 5 – sintesi PS



$$Z = (a + b' + c')(b' + c + e')(b + d' + e')(b + c' + d')$$

Esercizio 5 – sintesi NAND e NOR

$$Z = b'd' + abc + c'e'$$

$$Z_{NAND} = (b' \uparrow d') \uparrow (a \uparrow b \uparrow c) \uparrow (c' \uparrow e')$$

$$Z = (a + b' + c')(b' + c + e')(b + d' + e')(b + c' + d')$$

$$Z_{NOR} = (a \downarrow b' \downarrow c') \downarrow (b' \downarrow c \downarrow e') \downarrow (b \downarrow d' \downarrow e') \downarrow (b \downarrow c' \downarrow d')$$

Esercizio 6

L'uscita **Z** è l'uscita di un MUX 2:4 pilotato dai segnali **d** ed **e**.

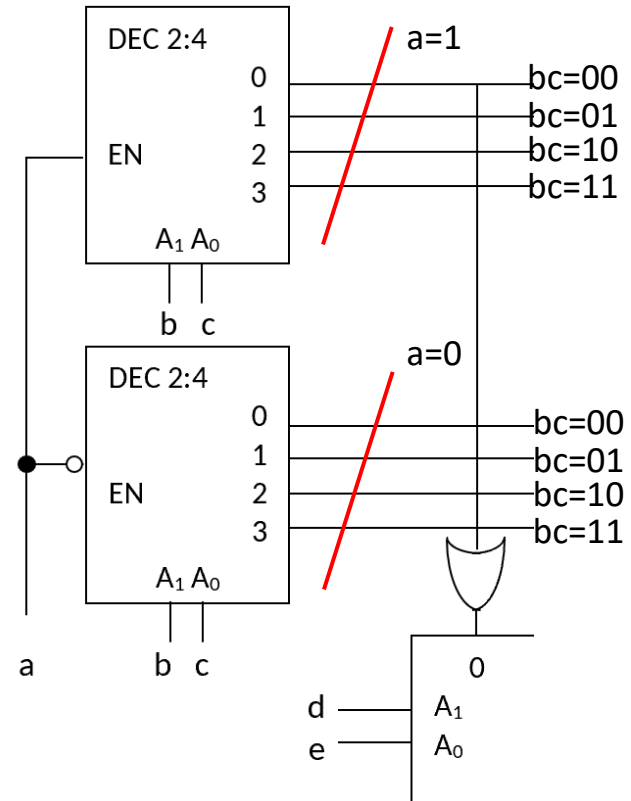
Quindi, per ogni combinazione di **d** ed **e**, la funzione in ingresso alla via selezionata dal valore di **d** ed **e** piloterà l'uscita Z.

Prendiamo il caso **de = 00** che corrisponde alla «colonna» di possibili connessioni collegata alla via 0 del MUX. In ingresso, ho una funzione di **a,b,c** da realizzare con sintesi DEC+OR.

Invece di un DEC 3:8, ho due DEC 2:4 in cui **a** attiva uno dei due decoder in base al suo valore. Questo circuito è equivalente ad un DEC 3:8 senza enable, con **a** bit di indirizzo, in quanto una delle prime 4 uscite potrà essere attiva solo quando **a=1**, e le altre solo quando **a=0**.

a	b	c	d	e	z
0	0	0	0	0	-
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	-
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	-
0	1	0	0	0	-
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	-
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	z
1	0	0	0	0	1
1	0	0	0	1	-
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	-
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	-
1	1	0	0	1	0
1	1	0	1	0	-
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	-
1	1	1	1	1	1



 Configurazioni d'ingresso con de=00

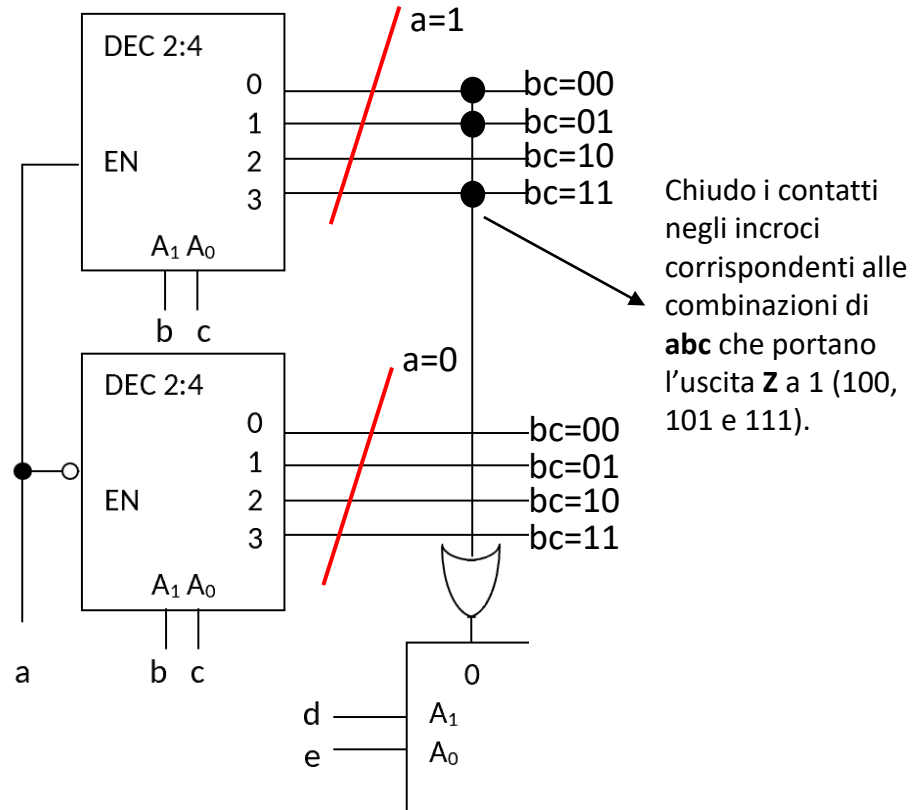
Esercizio 6

Nella sintesi DEC+OR mi interessano i mintermini, ovvero le configurazioni di **a,b,c** per cui l'uscita deve valere 1.

Devo quindi trovare le combinazioni di **a, b** e **c** con **de=00** che portano l'uscita **Z** ad 1. Guardo la tabella della verità:

a	b	c	d	e	z
0	0	0	0	0	-
0	0	0	0	1	1
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	-
0	0	1	0	1	1
0	0	1	1	0	0
0	0	1	1	1	-
0	1	0	0	0	-
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	-
0	1	1	0	0	0
0	1	1	0	1	0
0	1	1	1	0	0
0	1	1	1	1	0

a	b	c	d	e	z
1	0	0	0	0	1
1	0	0	0	1	-
1	0	0	1	0	1
1	0	0	1	1	0
1	0	1	0	0	1
1	0	1	0	1	-
1	0	1	1	0	0
1	0	1	1	1	0
1	1	0	0	0	-
1	1	0	0	1	0
1	1	0	1	0	-
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	-
1	1	1	1	1	1

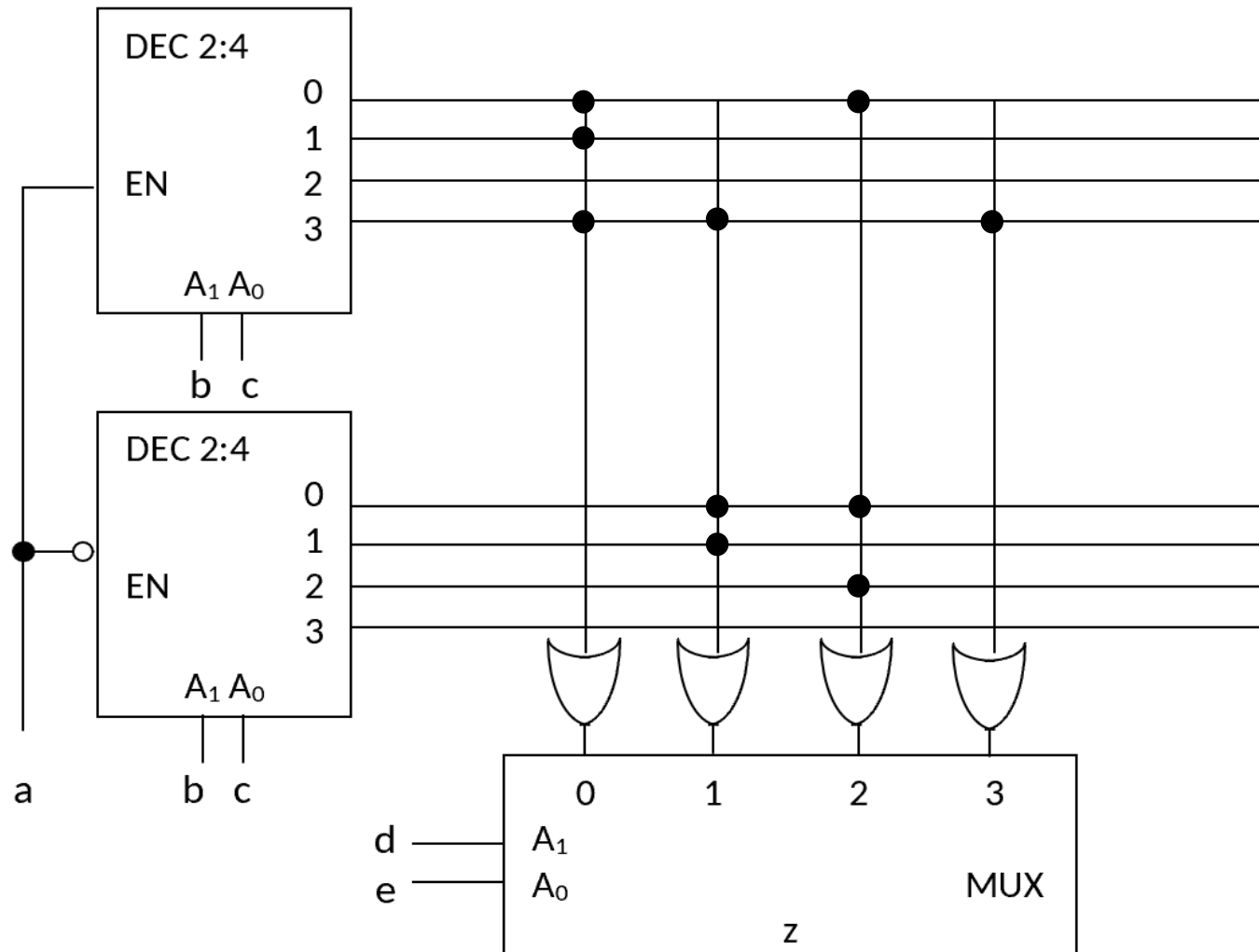


Configurazioni d'ingresso con de=00

Configurazioni d'ingresso con de=00 ed uscita Z=1

Esercizio 6

Ripetendo il ragionamento per tutte le configurazioni di **de** ottengo la soluzione completa.



Esercizio 7 – sintesi SP

x_1x_0 x_3x_2	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	0	1	1	1
10	0	1	1	1

$x_4 = 0$

x_1x_0 x_3x_2	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$x_4 = 1$

$$Z = x_4'x_3x_0 + x_4'x_1x_0' + x_4x_3'x_2' + x_4x_2x_0$$

Esercizio 7 – sintesi PS

x_1x_0 x_3x_2	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	0	1	1	1
10	0	1	1	1

$x_4 = 0$

x_1x_0 x_3x_2	00	01	11	10
00	1	1	1	1
01	0	1	1	0
11	0	1	1	0
10	0	0	0	0

$x_4 = 1$

$$Z = (x_4 + x_1 + x_0)(x_4 + x_3 + x'_0)(x'_4 + x'_3 + x_2)(x'_4 + x'_2 + x_0)$$

Esercizio 8 – sintesi SP

x_3	x_2	x_1	z_1	z_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	-	-
1	0	0	1	1
1	0	1	-	-
1	1	0	-	-
1	1	1	-	-



Configurazioni proibite



$x_3 \backslash x_2 x_1$	00	01	11	10
0	0	0	-	1
1	1	-	-	-

$$z_1 = x_3 + x_2$$

$x_3 \backslash x_2 x_1$	00	01	11	10
0	0	1	-	0
1	1	-	-	-

$$z_0 = x_3 + x_1$$

Esercizio 8 – analisi uscite per config. proibite

x_2x_1 x_3	00	01	11	10
0	0	0	-	1
1	1	-	-	-

$$z_1 = x_3 + x_2$$

x_2x_1 x_3	00	01	11	10
0	0	1	-	0
1	1	-	-	-

$$z_0 = x_3 + x_1$$

Le indifferenze all'interno dei RR diventano 1 sulle uscite!

x_3	x_2	x_1	z_1	z_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

 Configurazioni proibite

Esercizio 8 – sintesi SP con priorità

x_3	x_2	x_1	z_1	z_0
0	0	0	0	0
0	0	1	0	1
0	1	0	1	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

NO, priorità deve essere a x_2 . Le uscite devono valere 10.

OK, priorità a x_3 .

 Configurazioni proibite

$x_3 \backslash x_2 x_1$	00	01	11	10
0	0	0	1	1
1	1	1	1	1

$$z_1 = x_3 + x_2$$

$x_3 \backslash x_2 x_1$	00	01	11	10
0	0	1	0	0
1	1	1	1	1

$$z_0 = x_3 + x'_2 x_1$$

Esercizio 9 – sintesi SP

$de \backslash bc$	00	01	11	10
00	-	0	-	1
01	0	0	1	0
11	-	0	0	-
10	1	1	0	0

$a = 0$

$de \backslash bc$	00	01	11	10
00	1	1	1	-
01	-	-	0	-
11	-	0	0	1
10	1	0	0	-

$a = 1$

$$z = a'bc'd' + a'b'de + b'c'e' + ae' + ab'c'$$

Esercizio 9 – sintesi PS

$de \backslash bc$	00	01	11	10
00	-	0	-	1
01	0	0	1	0
11	-	0	0	-
10	1	1	0	0

$a = 0$

$de \backslash bc$	00	01	11	10
00	1	1	1	-
01	-	-	0	-
11	-	0	0	1
10	1	0	0	-

$a = 1$

$$z = (a + b + d)(a + b' + d')(a + b' + c')(a + c' + e)(a' + b' + e')(a' + b + c')$$

Esercizio 10 - TdV

Differenza

<i>s/d'</i>	<i>a</i>	<i>b</i>	<i>cbi</i>	<i>u</i>	<i>cbo</i>
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	1	0	1
0	0	1	0	1	1
0	1	0	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
0	1	1	0	0	0

Somma

<i>s/d'</i>	<i>a</i>	<i>b</i>	<i>cbi</i>	<i>u</i>	<i>cbo</i>
1	0	0	0	0	0
1	0	0	1	1	0
1	0	1	1	0	1
1	0	1	0	1	0
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	1	1	1
1	1	1	0	0	1

Esercizio 10 – sintesi SP

$\begin{array}{c} b\ cbi \\ s/d' a \end{array}$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	1	0	1	0
10	0	1	0	1

u

Attenzione alla notazione, si è usato \overline{x} per indicare la versione negata di un segnale per non fare confusione con il fatto che s/d' presenta il simbolo ' nel nome.

$$u = a \cdot \overline{b} \cdot \overline{cbi} + a \cdot b \cdot cbi + \overline{a} \cdot \overline{b} \cdot cbi + \overline{a} \cdot b \cdot \overline{cbi}$$

Esercizio 10 – sintesi SP

$\begin{matrix} b\ cbi \\ s/d' a \end{matrix}$	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

cbo

Attenzione alla notazione, si è usato \overline{x} per indicare la versione negata di un segnale per non fare confusione con il fatto che s/d' presenta il simbolo ' nel nome.

$$cbo = b \cdot cbi + \overline{s \backslash d'} \cdot \overline{a} \cdot cbi + \overline{s \backslash d'} \cdot \overline{a} \cdot b + s \backslash d' \cdot a \cdot cbi + s \backslash d' \cdot a \cdot b$$

Esercizio 10 – sintesi PS

$\begin{array}{c} b\ cbi \\ s/d' a \end{array}$	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	1	0	1	0
10	0	1	0	1

u

Attenzione alla notazione, si è usato \overline{x} per indicare la versione negata di un segnale per non fare confusione con il fatto che s/d' presenta il simbolo ' nel nome.

$$u = (\overline{a} + b + \overline{c}bi)(\overline{a} + \overline{b} + cbi)(a + b + cbi)(a + \overline{b} + \overline{c}bi)$$

Esercizio 10 – sintesi PS

$\begin{matrix} b\ cbi \\ s/d'\ a \end{matrix}$	00	01	11	10
00	0	1	1	1
01	0	0	1	0
11	0	1	1	1
10	0	0	1	0

cbo

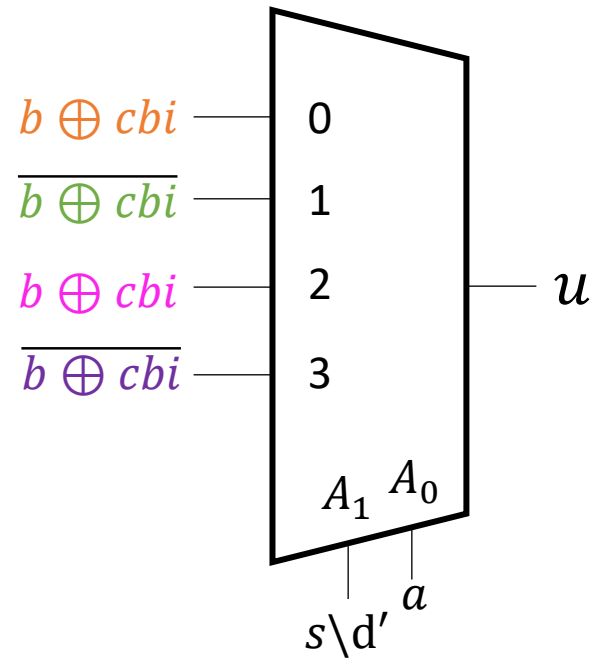
Attenzione alla notazione, si è usato \overline{x} per indicare la versione negata di un segnale per non fare confusione con il fatto che s/d' presenta il simbolo ' nel nome.

$$cbo = (b + cbi)(s \backslash d' + \overline{a} + b)(\overline{s \backslash d'} + a + cbi)(\overline{s \backslash d'} + a + b)(s \backslash d' + \overline{a} + cbi)$$

Esercizio 10 – sintesi con mux a 4 vie

Uso $s \backslash d'$ ed a come bit di indirizzo ed esprimo le uscite come funzione degli altri ingressi (b e cbi).

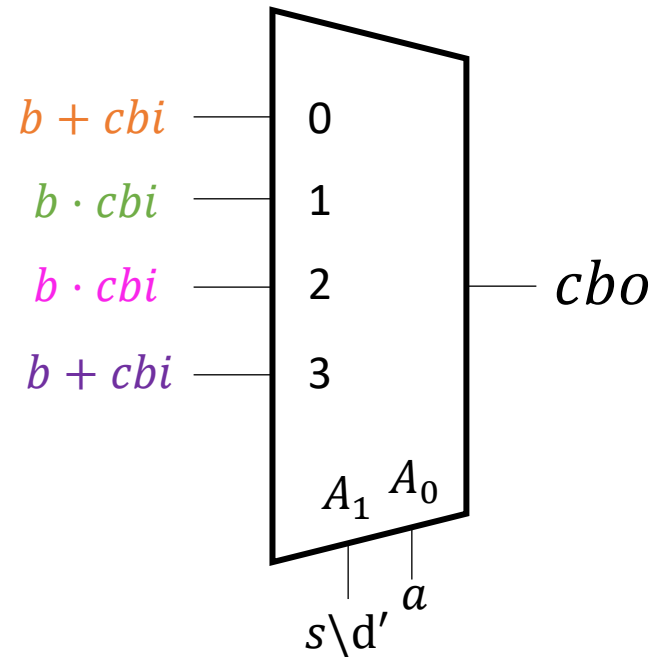
	$s \backslash d'$	a	b	cbi	u	cbo
$s \backslash d' e a = 00$	0	0	0	0	0	0
	0	0	0	1	1	1
	0	0	1	1	0	1
	0	0	1	0	1	1
$s \backslash d' e a = 01$	0	1	0	0	1	0
	0	1	0	1	0	0
	0	1	1	1	1	1
	0	1	1	0	0	0
	$s \backslash d'$	a	b	cbi	u	cbo
$s \backslash d' e a = 10$	1	0	0	0	0	0
	1	0	0	1	1	0
	1	0	1	1	0	1
	1	0	1	0	1	0
$s \backslash d' e a = 11$	1	1	0	0	1	0
	1	1	0	1	0	1
	1	1	1	1	1	1
	1	1	1	0	0	1



Esercizio 10 – sintesi con mux a 4 vie

Uso $s \backslash d'$ ed a come bit di indirizzo ed esprimo le uscite come funzione degli altri ingressi (b e cbi).

	s/d'	a	b	cbi	u	cbo
$s \backslash d' e a = 00$	0	0	0	0	0	0
	0	0	0	1	1	1
	0	0	1	1	0	1
	0	0	1	0	1	1
$s \backslash d' e a = 01$	0	1	0	0	1	0
	0	1	0	1	0	0
	0	1	1	1	1	1
	0	1	1	0	0	0
	s/d'	a	b	cbi	u	cbo
$s \backslash d' e a = 10$	1	0	0	0	0	0
	1	0	0	1	1	0
	1	0	1	1	0	1
	1	0	1	0	1	0
$s \backslash d' e a = 11$	1	1	0	0	1	0
	1	1	0	1	0	1
	1	1	1	1	1	1
	1	1	1	0	0	1

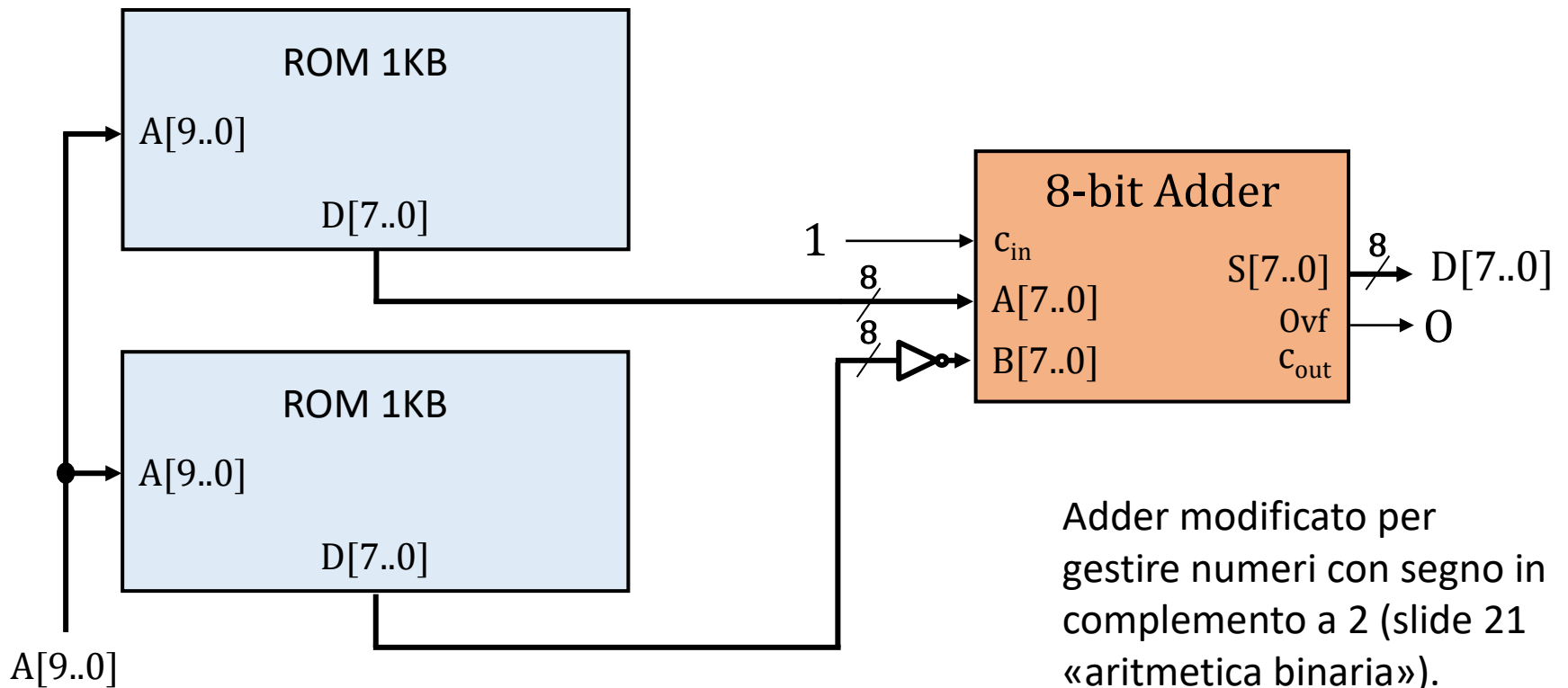


Esercizio 11

Una ROM da 1 KB richiede 10 bit di indirizzo $\rightarrow A[9..0]$

La differenza riportata sul bus D richiede 8 bit $\rightarrow D[7..0]$

La rete richiesta presenta 10 ingressi (i 10 bit che compongono $A[9..0]$) e 9 uscite (gli 8 bit che compongono $D[7..0]$ ed il segnale O).



Esercizio 12 - [-3, 1]

Valore	$N_3N_2N_1N_0$
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Intervallo: [-3, 1]

Codifica: complemento a 2

Numero bit: 4

$$Y = N_3N_2(N_1 + N_0) + N'_3N'_2N'_1$$

Da -3 a -1: N_3 e N_2 entrambi a 1
ed almeno uno tra N_1 ed N_0 a 1

Da 0 a 1: N_3, N_2 ed N_1 tutti a 0

Esercizio 12 – [41, 63]

Intervallo: [41, 63], Codifica: interi senza segno, Numero bit: 7

L'intervallo di numeri interi senza segno rappresentabili con 7 bit è [0, 127].

Il bit più significativo (N_6) «divide» l'intervallo in 2: quando vale 0 siamo nella metà [0, 63], quando vale 1 siamo nella metà [64, 127]. Quindi intanto sappiamo che siamo interessati alla metà « $N_6 = 0$ ». In questa metà, estendendo il ragionamento appena fatto, vediamo che $N_5 = 0$ ci porta nell'intervallo [0, 31], mentre $N_5 = 1$ ci porta nell'intervallo [32, 63]. Quindi sappiamo che siamo interessati all'intervallo $N_6 N_5 = 01$.

Vediamo nel dettaglio questo intervallo:

Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$
32	0100000	40	0101000	48	0110000	56	0111000
33	0100001	41	0101001	49	0110001	57	0111001
34	0100010	42	0101010	50	0110010	58	0111010
35	0100011	43	0101011	51	0110011	59	0111011
36	0100100	44	0101100	52	0110100	60	0111100
37	0100101	45	0101101	53	0110101	61	0111101
38	0100110	46	0101110	54	0110110	62	0111110
39	0100111	47	0101111	55	0110111	63	0111111

Esercizio 12 - [41, 63]

Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$
32	0100000	40	0101000	48	0110000	56	0111000
33	0100001	41	0101001	49	0110001	57	0111001
34	0100010	42	0101010	50	0110010	58	0111010
35	0100011	43	0101011	51	0110011	59	0111011
36	0100100	44	0101100	52	0110100	60	0111100
37	0100101	45	0101101	53	0110101	61	0111101
38	0100110	46	0101110	54	0110110	62	0111110
39	0100111	47	0101111	55	0110111	63	0111111

$$Y = N'_6N'_5N'_4N_3 \cdot (N_2 + N_1 + N_0) + N'_6N'_5N_4$$

$[41, 47] \rightarrow N_6 = 0, N_5 = 1, N_4 = 0, N_3 = 1$
ed almeno uno tra N_2, N_1 ed N_0 ad 1

$[48, 63] \rightarrow N_6 = 0, N_5 = 1, N_4 = 1$

Esercizio 12 – [-288, -1]

Intervallo: [-288, -1], Codifica: complemento a 2, Numero bit: 10

L'intervallo di numeri rappresentabili in complemento a 2 con 10 bit è [-512, 511].

L'intervallo a cui sono interessato comprende solo numeri negativi, quindi so che $N_9 = 1$.

$N_9 = 1$ identifica l'intervallo [-512, -1], da qui considero che $N_8 = 0$ identifica il sottointervallo [-512, -257] mentre $N_8 = 1$ identifica l'intervallo [-256, -1].

Con queste considerazioni posso capire che devo «spezzare» l'intervallo [-288, -1] in «[-288, -257] o [-256, -1]». Infatti, l'intervallo [-256, -1] può essere facilmente rappresentato con l'espressione $N_9 N_8$ (che corrisponde a $N_9 = 1$ e $N_8 = 1$).

Ragiono su come rappresentare l'intervallo [-288, -257]: siamo nell'intervallo [-512, -257], identificato da $N_9 = 1$ e $N_8 = 0$. In questo intervallo, $N_7 = 0$ identifica l'intervallo [-512, -385] mentre $N_7 = 1$ identifica l'intervallo [-384, -257]. Proseguendo il ragionamento, quando $N_7 = 1$, $N_6 = 0$ identifica l'intervallo [-384, -321] mentre $N_6 = 1$ identifica l'intervallo [-320, -257]. Infine, quando $N_7 = 1$ e $N_6 = 1$, $N_5 = 0$ identifica l'intervallo [-320, -289], mentre $N_5 = 1$ identifica l'intervallo [-288, -257], ossia quello che ci interessa. Ricapitolando, l'intervallo [-288, -257] è identificato da $N_9 = 1, N_8 = 0, N_7 = 1, N_6 = 1, N_5 = 1$, ossia dall'espressione $N_9 N_8' N_7 N_6 N_5$.

In conclusione:

$$Y = N_9 N_8' N_7 N_6 N_5 + N_9 N_8$$

[-288, -257] o [-256, -1]

Esercizio 12 – [15, 40]

Intervallo: [15, 40], Codifica: interi senza segno, Numero bit: 7

L'intervallo di numeri interi senza segno rappresentabili con 7 bit è [0, 127].

Il bit più significativo (N_6) «divide» l'intervallo in 2: quando vale 0 siamo nella metà [0, 63], quando vale 1 siamo nella metà [64, 127]. Quindi intanto sappiamo che siamo interessati alla metà « $N_6 = 0$ ». In questa metà, estendendo il ragionamento appena fatto, vediamo che $N_5 = 0$ ci porta nell'intervallo [0, 31], mentre $N_5 = 1$ ci porta nell'intervallo [32, 63].

Da queste considerazioni, capiamo che dobbiamo dividere l'intervallo che ci interessa in due, ossia [15, 31] e [32, 40].

Ragioniamo su [15, 31]: $N_6 = 0$ e $N_5 = 0$ identificano l'intervallo [0, 31]. In questo intervallo, $N_4 = 0$ ci porta nell'intervallo [0, 15] mentre $N_4 = 1$ ci porta nell'intervallo [16, 31]. Visto che 15 è l'estremo superiore di uno di questi due intervalli, dovrò prenderlo singolarmente con la sua codifica completa, spezzando l'intervallo [15, 31] in «15 o [16, 31]», rappresentabile con l'espressione $N'_6 N'_5 N'_4 N_3 N_2 N_1 N_0 + N'_6 N'_5 N_4$.

Ragioniamo ora su [32, 40]: $N_6 = 0$ e $N_5 = 1$ identificano l'intervallo [32, 63]. In questo intervallo, $N_4 = 0$ ci porta nell'intervallo [32, 47] mentre $N_4 = 1$ ci porta nell'intervallo [48, 63]. Vediamo nel dettaglio l'intervallo [32, 47]:

Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$	Valore	$N_6 N_5 N_4 N_3 N_2 N_1 N_0$
32	0100000	36	0100100	40	0101000	44	0101100
33	0100001	37	0100101	41	0101001	45	0101101
34	0100010	38	0100110	42	0101010	46	0101110
35	0100011	39	0100111	43	0101011	47	0101111

Esercizio 12 – [15, 40]

Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$	Valore	$N_6N_5N_4N_3N_2N_1N_0$
32	0100000	36	0100100	40	0101000	44	0101100
33	0100001	37	0100101	41	0101001	45	0101101
34	0100010	38	0100110	42	0101010	46	0101110
35	0100011	39	0100111	43	0101011	47	0101111

L'intervallo [32, 40] può quindi essere rappresentato come «[32, 39] o 40» con l'espressione:

$$N'_6N'_5N'_4N'_3 + N'_6N'_5N'_4N'_3N'_2N'_1N'_0$$

Aggregando quanto detto, l'intervallo [15, 40] può essere rappresentato dall'espressione logica «15 o [16, 31] o [32, 39] o 40»:

$$Y = N'_6N'_5N'_4N'_3N'_2N'_1N'_0 + N'_6N'_5N'_4 + N'_6N'_5N'_4N'_3 + N'_6N'_5N'_4N'_3N'_2N'_1N'_0$$